Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended): A system for efficiently forwarding client requests in a distributed computing environment, comprising:

a socket receiving a plurality of ~~non-proxiable~~ client requests commonly ~~addressed~~ routed for forwarding to an origin server from individual sending clients into a proxy server connectively interposed between the sending clients and the origin server;

a time estimates generator dynamically generating at the proxy server, concurrent to and during processing of each request by the proxy server, time estimates of service availability based on a time-to-idle for sending the requests over each of a plurality of network connections from the proxy server to the origin server, wherein time-to-idle for each network connection is calculated based on the amount of time that will elapse before an active network connection is usable for a subsequent client request; and

a network connection manager selecting the network connection from the proxy server to the origin server with a substantially highest service availability and a substantially lowest time-to-idle and forwarding each request from the proxy server to the origin server using the selected network connection.

2. (currently amended): A system according to Claim 1, further comprising:

the network connection manager selecting a network connection from the proxy server not actively sending a request with a zero time-to-idle and not subject to a slow start overhead incurred responsive to flow control imposed by the sending client.

3. (currently amended): A system according to Claim 2, further comprising:

the network connection manager selecting a network connection from the proxy server actively sending a request with a time-to-idle less than the slow start overhead, plus request transfer time if the network connection ~~is pipelined~~ allows request pipelining.

4. (previously presented): A system according to Claim 3, further comprising:

the network connection manager selecting a network connection not actively sending a request with a zero time-to-idle and subject to the slow start overhead.

5. (currently amended): A system according to Claim 4, further comprising:

the network connection manager selecting a network connection from the proxy server actively sending a request with a time-to-idle less than a network connection setup overhead, plus request transfer time if the network connection ~~is pipelined~~ allows request pipelining.

6. (currently amended): A system according to Claim 5, further comprising:

the network connection manager selecting a new network connection from the proxy server in the absence of an existing network connection with a time-to-idle less than the network connection setup overhead.

7. (currently amended): A system according to Claim 5, further comprising:

the network connection manager selecting an existing network connection from the proxy server with the substantially lowest time-to-idle.

8. (currently amended): A system according to Claim 1, wherein the distributed operating environment is TCP/IP-compliant, the system further comprising:

the time estimates generator providing time estimates for each network connection from the proxy server comprising at least one of TCP overhead, time-to-idle, idle time, and request transfer time, wherein idle time for each network connection is measured as an amount of time that has elapsed during which no network traffic is being sent or received over a network connection.

9. (previously presented): A system according to Claim 8, the network connection setup overhead comprises TCP overhead, the system further comprising:

the time estimates generator calculating the TCP overhead by adding a three-way handshake overhead to a slow start overhead.

10. (previously presented): A system according to Claim 8, further comprising:

the time estimates generator calculating the request transfer time by multiplying the size of the request by an average network connection speed for the origin server.

11. (previously presented): A system according to Claim 8, further comprising:

the time estimates generator calculating the time-to-idle upon each receipt of a request by adding the time-to-idle to the product of an average network connection speed for the origin server multiplied by the sum of the request size and an estimated response size.

12. (previously presented): A system according to Claim 8, further comprising:

the time estimates generator calculating the time-to-idle upon writing data to a socket by subtracting the time-to-idle from the product of an average network connection speed for the origin server multiplied by the amount of data written.

13. (previously presented): A system according to Claim 8, further comprising:

the time estimates generator calculating the time-to-idle upon reading data from a socket, prior to header data, by subtracting the time-to-idle from the product of an average network connection speed for the origin server multiplied by the amount of data read.

14. (currently amended): A system according to Claim 1, further comprising:

[[~~a~~]] one such proxy server configured in a location comprising at least one of local to the sending clients, in the infrastructure of the distributed computing environment, and local to the origin server.

15. (currently amended): A method for efficiently forwarding client requests in a distributed computing environment, comprising:

receiving a plurality of ~~non-proxiable~~ client requests commonly ~~addressed~~ routed for forwarding to an origin server from individual sending clients into a proxy server connectively interposed between the sending clients and the origin server;

dynamically generating at the proxy server, concurrent to and during processing of each request by the proxy server, time estimates of service availability based on a time-to-idle for sending the requests over each of a plurality of network connections from the proxy server to the origin server, wherein time-to-idle for each network connection is calculated based on the amount of time that will elapse before an active network connection is usable for a subsequent client request; and

selecting the network connection from the proxy server to the origin server with a substantially highest service availability and a substantially lowest time-to-idle and forwarding each request from the proxy server to the origin server using the selected network connection.

16. (currently amended): A method according to Claim 15, further comprising:

selecting a network connection from the proxy server not actively sending a request with a zero time-to-idle and not subject to a slow start overhead incurred responsive to flow control imposed by the sending client.

17. (currently amended): A method according to Claim 16, further comprising:

selecting a network connection from the proxy server actively sending a request with a time-to-idle less than the slow start overhead, plus request transfer time if the network connection ~~is pipelined~~ allows request pipelining.

18. (previously presented): A method according to Claim 17, further comprising:

selecting a network connection not actively sending a request with a zero time-to-idle and subject to the slow start overhead.

19. (currently amended): A method according to Claim 18, further comprising:

selecting a network connection from the proxy server actively sending a request with a time-to-idle less than a network connection setup overhead, plus request transfer time if the network connection ~~is pipelined~~ allows request pipelining.

20. (currently amended): A method according to Claim 19, further comprising:

selecting a new network connection from the proxy server in the absence of an existing network connection with a time-to-idle less than the network connection setup overhead.

21. (currently amended): A method according to Claim 19, further comprising:

selecting an existing network connection with from the proxy server the substantially lowest time-to-idle.

22. (currently amended): A method according to Claim 15, wherein the distributed operating environment is TCP/IP-compliant, the method further comprising:

providing time estimates for each network connection from the proxy server comprising at least one of TCP overhead, time-to-idle, idle time, and request transfer time, wherein idle time for each network connection is measured as an amount of time that has elapsed during which no network traffic is being sent or received over a network connection.

23. (previously presented): A method according to Claim 22, the network connection setup overhead comprises TCP overhead, the method further comprising:

calculating the TCP overhead by adding a three-way handshake overhead to a slow start overhead.

24. (previously presented): A method according to Claim 22, further comprising:

calculating the request transfer time by multiplying the size of the request by an average network connection speed for the origin server.

25. (previously presented): A method according to Claim 22, further comprising:

calculating the time-to-idle upon each receipt of a request by adding the time-to-idle to the product of an average network connection speed for the origin server multiplied by the sum of the request size and an estimated response size.

26. (previously presented): A method according to Claim 22, further comprising:

calculating the time-to-idle upon writing data to a socket by subtracting the time-to-idle from the product of an average network connection speed for the origin server multiplied by the amount of data written.

27. (previously presented): A method according to Claim 22, further comprising:

calculating the time-to-idle upon reading data from a socket, prior to header data, by subtracting the time-to-idle from the product of an average network connection speed for the origin server multiplied by the amount of data read.

28. (currently amended): A method according to Claim 15, further comprising:

providing [[~~a~~]] one such proxy server configured in a location comprising at least one of local to the sending clients, in the infrastructure of the distributed computing environment, and local to the origin server.

29. (original): A computer-readable storage medium holding code for performing the method according to Claim 15.

30. (currently amended): A system for efficiently forwarding client requests from a proxy server in a TCP/IP computing environment, comprising:

means for receiving on a proxy server connectively interposed between the sending clients and an origin server a plurality of transient requests from individual sending clients, each request being commonly ~~addressed~~ routed for forwarding to [[~~an~~]] the origin server;

means for dynamically calculating at the proxy server, concurrent to receiving and during processing of each request, time estimates of TCP overhead, slow start overhead, time-to-idle, and request transfer time for sending the requests over each of a plurality of managed network connections from the proxy server to the origin server, wherein time-to-idle for each network connection is calculated based on the amount of time that will elapse before an active network connection is usable for a subsequent request;

means for choosing the managed network connection from the proxy server selected from, in order of preferred selection, a warm idle network connection, an active network connection with a time-to-idle less than a slow start overhead, a cold idle network connection, an active network connection with a time-to-idle less than a TCP overhead, a new managed network connection, and an existing managed network connection with a smallest time-to-idle; and

means for forwarding each request from the proxy server to the origin server over the selected managed network connection.

31. (currently amended): A system according to Claim 30, further comprising:

means for adding the request transfer time during each active network connection selection if the managed network connection ~~is pipelined~~ from the proxy server allows request pipelining.

32. (previously presented): A system according to Claim 30, further comprising:

means for calculating the TCP overhead by adding a three-way handshake overhead to a slow start overhead;

means for calculating the request transfer time by multiplying the size of the request by an average managed network connection speed for the origin server; and

means for calculating the time-to-idle, comprising:

upon each receipt of a request, means for adding the time-to-idle to the product of an average managed network connection speed for the origin server multiplied by the sum of the request size and an estimated response size;

upon writing data to a socket, means for subtracting the time-to-idle from the product of an average managed network connection speed for the origin server multiplied by the amount of data written; and

upon reading data from a socket, prior to header data, means for subtracting the time-to-idle from the product of an average managed network connection speed for the origin server multiplied by the amount of data read.

33. (original): A system according to Claim 30, wherein each transient request is communicated in accordance with HTTP.

34. (currently amended): A method for efficiently forwarding client requests from a proxy server in a TCP/IP computing environment, comprising:

receiving a plurality of transient requests from individual sending clients into a request queue on a proxy server connectively interposed between the sending clients and an origin server, each request being commonly ~~addressed~~ routed for forwarding to [[~~an~~]] the origin server;

dynamically calculating at the proxy server, concurrent to receiving and during processing of each request, time estimates of TCP overhead, slow start overhead, time-to-idle, and request transfer time for sending the requests over each of a plurality of managed network connections from the proxy server to the origin server, wherein time-to-idle for each network connection is calculated based on the amount of time that will elapse before an active network connection is usable for a subsequent request;

choosing the managed network connection from the proxy server selected from, in order of preferred selection, a warm idle network connection, an active network connection with a time-to-idle less than a slow start overhead, a cold idle network connection, an active network connection with a time-to-idle less than a TCP overhead, a new managed network connection, and an existing managed network connection with a smallest time-to-idle; and

forwarding each request from the proxy server to the origin server over the selected managed network connection.

35. (currently amended): A method according to Claim 34, further comprising:

adding the request transfer time during each active network connection selection if the managed network connection ~~is pipelined~~ from the proxy server allows request pipelining.

36. (previously presented): A method according to Claim 34, further comprising:

calculating the TCP overhead by adding a three-way handshake overhead to a slow start overhead;

calculating the request transfer time by multiplying the size of the request by an average managed network connection speed for the origin server; and

calculating the time-to-idle, comprising:

upon each receipt of a request, adding the time-to-idle to the product of an average managed network connection speed for the origin server multiplied by the sum of the request size and an estimated response size;

upon writing data to a socket, subtracting the time-to-idle from the product of an average managed network connection speed for the origin server multiplied by the amount of data written; and

upon reading data from a socket, prior to header data, subtracting the time-to-idle from the product of an average managed network connection speed for the origin server multiplied by the amount of data read.

37. (original): A method according to Claim 34, wherein each transient request is communicated in accordance with HTTP.

38. (original): A computer-readable storage medium holding code for performing the method according to Claim 34.